UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/734,390	12/12/2003	Ramin Oliver Assadollahi	Assadollahi PIM	9756

33525 7590 02/15/2011
JONATHAN D. FEUCHTWANG
2912 BRITTAN AVE
SAN CARLOS, CA 94070

EXAMINER	
ALVESTEFFER, STEPHEN D	

ART UNIT	PAPER NUMBER
2175	

MAIL DATE	DELIVERY MODE
02/15/2011	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/734,390	ASSADOLLAHI, RAMIN OLIVER	
	Examiner	Art Unit	
	Stephen Alvesteffer	2175	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *30 September 2010*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1,3-8,10 and 13-42* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1,3-8,10 and 13-42* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

Response to Amendment

This Office Action is responsive to the Request for Continued Examination (RCE) filed September 30, 2010. Claims 1 and 10 are amended. Claims 2, 11, and 12 are currently cancelled. Claim 9 was previously cancelled. Claims 13-42 are new. Claims 1, 10, and 13 are independent. Claims 1, 3-8, 10, and 13-42 remain pending.

Claim Objections

Claims 5, 10, 17, 20, 23, 26, 34, 37, and 39 are objected to because of the following informalities:

In claim 5 line 7, "characteristic word used refine the data processing request" should be corrected to –characteristic word used to refine the data processing request—

In claim 10 line 7, "being associated at least one argument" should be corrected to –being associated with at least one argument—.

In claim 17 line 2, "to/from the least one dedicated database" should be corrected to –to/from the at least one dedicated database—.

In claim 20 lines 1-2, "stored in the least one dedicated database" should be corrected to –stored in the at least one dedicated database—.

In claim 23 line 5, "word used refine the data" should be corrected to –word used to refine the data—.

In claim 26 line 4, “provided that that the selected word" should be corrected to –provided that ~~that~~ the selected word—.

In claim 34 line 2, “to/from the least one dedicated database” should be corrected to –to/from the at least one dedicated database—.

In claim 37 lines 1-2, “stored in the least one dedicated database” should be corrected to –to/from the at least one dedicated database—.

In claim 39 line 5, “word used refine the data” should be corrected to –word used to refine the data—.

Appropriate correction is required.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Claims 1 and 6 are rejected under 35 U.S.C. 103(a) as being unpatentable over Mattson et al. (hereinafter Mattson), United States Patent 5,303,148, and Fein et al. (hereinafter Fein), United States Patent 6,012,075.

Regarding claim 1, Mattson substantially teaches a personal information manager comprising:

a microprocessor (see Mattson Abstract; “*speech processor*”);

memory operably connected to the microprocessor and storing a database (see Mattson column 4 lines 11-34; "*the system manager 16 may enable the speech processor and a text memory means 26 to transform the verbal observations to text or words and store words generated*");

a data input device operably connected to the microprocessor and configured to receive an audio data stream and decode the received audio data stream into text (see Mattson column 4 lines 11-34; "*the physician's observations are converted from the spoken word to text*");

said database storing decoded text, and a table of explicit commands (see Mattson column 3 lines 34-50, "*A speech processor 12 analyzes the received audio signal and generates corresponding electronic words or text, as is known in the art. A command interpreter 14 compares each generated word or text with a list of preselected control commands or command words. The list of preselected control commands includes all or some of the commands which the operator had previously been able to enter on the keyboard C.*");

a dialog manager module executed by the microprocessor and having a record mode and a dialog mode, in said record mode said dialog manager comparing said decoded text received from said data input device with said table of explicit commands to determine whether it contains a request explicitly requested by a user and which is immediately passed to the microprocessor for execution (see Mattson column 3 lines 34-50, "*A command interpreter 14 compares each generated word or text with a list of preselected control commands or command words. The list of preselected control*

commands includes all or some of the commands which the operator had previously been able to enter on the keyboard C. In response to receiving each of the preselected command words, the command interpreter generates the same control signal that the keyboard c produces for the corresponding command. The control input signal is processed by the system manager 16 which is responsible for activating the appropriate response of the whole system.");

an information storage/retrieval module executed by the microprocessor and storing and retrieving text to/from said database, said information storage/retrieval module handling implicit and explicit data processing requests specified by said dialog manager, in said record mode and dialog manager module instructing the information storage/retrieval module to store decoded text, excluding explicit data processing requests, in said database (see Mattson column 4 lines 11-34; "*the system manager 16 may enable the speech processor and a text memory means 26 to transform the verbal observations to text or words and store words generated*");

said dialog manager module examining text stored in said database during periods of microprocessor inactivity to determine whether the stored text contains implicit data processing requests, where implicit processing requests are determined by examining a semantic class of the stored text, said dialog manager module adding implicit processing requests to an implicit processing queue and executing implicit processing requests during periods of microprocessor inactivity (Fein, addressed below); and

an output module converting text received from said dialog manager module into speech and outputting said speech in response to a data processing request (see Mattson column 5 lines 3-23; "*The speech synthesizer 42 is connected with the memory means 22 to convert the words of the stored patient history into audible signals to be supplied to a speaker 44*").

Mattson does not teach every limitation of said dialog manager module examining text stored in said database during periods of microprocessor inactivity to determine whether the stored text contains implicit data processing requests, where implicit processing requests are determined by examining a semantic class of the stored text, said dialog manager module adding implicit processing requests to an implicit processing queue and executing implicit processing requests during periods of microprocessor inactivity. Fein teaches a method and system that performs implicit grammar checking of stored text during periods of microprocessor inactivity (see Fein column 4 lines 48-67, "*The preferred program includes a grammar checker program module. During idle periods, i.e., while a user is not entering data or commands, the document is automatically grammar checked in the background. The user does not have to enter any commands to initiate the background grammar check.*"; see Fein column 9 lines 19-29, "*The executable file 200 further includes an idle task manager 220 which manages tasks performed in the background. "Background" in this context means that the preferred application program module is idle, such as when the user is not typing and when no other commands are being executed. During an idle period, the idle task manager determines which idle tasks need to be performed based on a priority*

scheme. Those skilled in the art will be familiar with the operations and tasks performed by an idle task manager, including managing the order of performance of idle tasks during an idle period based on a priority scheme."; see also Fein column 16 lines 4-14, "*After marking the sentence with a grammarclean flag at step 535 or marking the characters comprising an error with a grammarerror flag at step 540, the method proceeds to decision step 545. A determination is made whether there are any more grammardirty flags or out-of-date grammarerror flags in the text run at decision step 545. If it is determined that there are no more grammardirty flags or out-of-date grammarerror flags in the text run being checked, then the method ends at step 517 and the preferred application program receives instruction from the idle task manager as to the next idle task to perform.*"). It would have been obvious to one having ordinary skill in the art at the time the invention was made to perform implicit tasks on the stored text during periods of microprocessor inactivity as taught by Fein in the invention of Mattson so that processing or analysis may be performed on the stored text without affecting the performance of the system.

Regarding claim 6, Mattson/Fein teaches a global word table containing a list of all of the words stored in the database (see Mattson column 4 lines 35-54; "*A comparing means 34 compares the digitized audio signal or word with a library 36 of stored digital words*"); and

said dialog manager module examining decoded text received from said data input device to determine whether it matches to a given said word in said global word

table (see Mattson column 4 lines 35-54; “*A comparing means 34 compares the digitized audio signal or word with a library 36 of stored digital words*”);

wherein a request to prompt the user for clarification is queued if the decoded text does not match any word in said global word table (see Mattson column 4 lines 35-54; "*To initialize the memory, each operator is shown each command and asked to vocalize an audio input corresponding thereto*").

Claims 3-5, 7, 8, and 10 are rejected under 35 U.S.C. 103(a) as being unpatentable over Mattson (US 5,303,148) *supra*, Fein (US 6,012,075) *supra*, and Dunning, United States Patent 7,162,482.

Regarding claim 3, Mattson/Fein teaches every limitation of claim 3 except that said dialog manager module identifies an explicit data processing request during said dialog mode by comparing said decoded text against a list of predefined data processing requests, assigning a match score to each of said predefined data processing requests and selecting said predefined data processing request having a highest matching score as said explicit data processing request. Mattson teaches comparing the digitized signal or word with a library of stored digital words (see Mattson column 4 lines 35-54), but does not explicitly disclose how this is done internally. Dunning explicitly teaches determining a match score to determine if an input matches a stored value (see Dunning column 9 lines 7-13; “*Once an index has been built it can be used to identify an unknown signal. The unknown signal can also be broken into documents, quantized, and grouped into words. In one embodiment of the present*

invention, the words in the unknown documents can be compared to the words in the known documents in order to find a match and identify an unknown piece of music by its tag information."). It would have been obvious to one having ordinary skill in the art at the time the invention was made to compare input to stored values as taught by Dunning in the invention of Mattson/Fein in order to properly recognize if spoken commands match reserved words stored in the database.

Regarding claim 4, Mattson/Fein/Dunning teaches that if said highest matching score is less than a threshold score said dialog manager module passes an instruction to said output module to prompt the user to select a given data processing request from among a selected number of said predefined data processing requests (see Dunning Figure 8 and column 9 lines 14-27; "*Now referring to FIG. 8, there is shown a method of forming a word from letters. Words are formed from a series of letters in a given document. In one embodiment, there are two thresholds that together define whether a word is considered "frequently appearing" in each document. The threshold values are chosen such that the words yield an accurate, fast, and memory-efficient result of identifying an unknown signal. The first threshold is a minimum number of appearances of a word in a document. The first threshold is referred to as t.sub.1. In one embodiment, the second threshold is a maximum number of appearances of a word in a document. The second threshold is referred to as t.sub.2. A word is considered to be "frequently appearing" if its frequency lies between the thresholds. In an alternate embodiment, only one of the two thresholds is used.*"). Figure 8 shows a method of forming a word from letters. Words are formed from a series of letters in a given

document. In one embodiment, there are two thresholds that together define whether a word is considered “frequently appearing” in each document. The threshold values are chosen such that the words yield an accurate, fast, and memory-efficient result of identifying an unknown signal. The first threshold is a minimum number of appearances of a word in a document. The first threshold is referred to as t.sub.1. In one embodiment, the second threshold is a maximum number of appearances of a word in a document. The second threshold is referred to as t.sub.2. A word is considered to be “frequently appearing” if its frequency lies between the thresholds. Furthermore, depending on the threshold letter grouping is provided to the user. It would have been obvious to one of ordinary skill in the art at the time of the invention to combine Mattson’s invention with Dunning’s threshold aspect because this would allow to effective separability and prediction.

Regarding claim 5, Mattson/Fein/Dunning teaches that said information storage/retrieval module passes to said dialog manager module a specified number of data records retrieved in response to said data processing request if a number of retrieved data records is below a threshold number and otherwise passes characteristic words selected from said retrieved data records, and said dialog manager module instructs said output module to prompt the user to select a given said characteristic word used refine the data processing request (see Dunning Figure 8 and column 9 lines 14-27; “*Now referring to FIG. 8, there is shown a method of forming a word from letters. Words are formed from a series of letters in a given document. In one embodiment, there are two thresholds that together define whether a word is considered "frequently*

appearing" in each document. The threshold values are chosen such that the words yield an accurate, fast, and memory-efficient result of identifying an unknown signal. The first threshold is a minimum number of appearances of a word in a document. The first threshold is referred to as t.sub.1. In one embodiment, the second threshold is a maximum number of appearances of a word in a document. The second threshold is referred to as t.sub.2. A word is considered to be "frequently appearing" if its frequency lies between the thresholds. In an alternate embodiment, only one of the two thresholds is used."). Applicant should duly note that it is well known in the art at the time of the invention to provide a listing of results even if the data record retrieved is below the threshold rather than saying the information requested was not found.

Regarding claim 7, Mattson/Fein/Dunning teaches a local word table in said database (see Mattson column 4 lines 35-54; "*A comparing means 34 compares the digitized audio signal or word with a library 36 of stored digital words*");

said information storage/retrieval module stores atoms of data, each said atom having a unique identifier (see Mattson column 4 lines 35-54; "*The library 36 includes a plurality of digitized word forms corresponding to each selectable command*"); and

said local word table containing a list of words contained in each atom of data and the number of times each word appears in a given atom (see Dunning Figures 1 and 8, wherein words are formed based on the frequency of the groups of letters);

wherein if a number of atoms matching a data retrieval request exceeds a predetermined number, said dialog manager module prompts a user to select a given characteristic word from a list of characteristic words, said characteristic words being

derived from the local word tables of atoms matching said data retrieval request, said selected characteristic word being appended to a search string of the data retrieval request, thereby reducing the number of atoms matching a data retrieval request (see Dunning column 9 lines 7-13; "*Once an index has been built it can be used to identify an unknown signal. The unknown signal can also be broken into documents, quantized, and grouped into words. In one embodiment of the present invention, the words in the unknown documents can be compared to the words in the known documents in order to find a match and identify an unknown piece of music by its tag information.*", if a letter or group of letters appears within certain threshold frequency limits, then the letter or group of letters are organized to form words. Once an index has been built it can be used to identify an unknown signal. The unknown signal can also be broken into documents, quantized, and grouped into words. Furthermore, the words in the unknown documents can be compared to the words in the known documents in order to find a match and identify an unknown piece of music by its tag information). Dunning inherently discloses the claimed aspect of atomization of data content. Even if not disclosed inherently Vethe in 5,991,765 discloses the claimed aspect of atomization of data and furthermore linking of atoms to create more complex data in the database. Applicant should duly note that it is well known in the art at the time the invention was made to show a predetermined number of results and if the number of search results exceeds the predetermined number an additional criteria is given to further refine the search result.

Regarding claim 8, Mattson/Fein/Dunning teaches that said characteristic words are derived by selecting a predetermined number of the most frequently occurring

words from the local word tables of the atoms matching a data retrieval request, provided that the selected word does not already appear in the search string of the data retrieval request (see Dunning Figure 8 and column 9 lines 14-27; “*Now referring to FIG. 8, there is shown a method of forming a word from letters. Words are formed from a series of letters in a given document. In one embodiment, there are two thresholds that together define whether a word is considered "frequently appearing" in each document. The threshold values are chosen such that the words yield an accurate, fast, and memory-efficient result of identifying an unknown signal. The first threshold is a minimum number of appearances of a word in a document. The first threshold is referred to as t.sub.1. In one embodiment, the second threshold is a maximum number of appearances of a word in a document. The second threshold is referred to as t.sub.2. A word is considered to be "frequently appearing" if its frequency lies between the thresholds. In an alternate embodiment, only one of the two thresholds is used.*”). Applicant should duly note that the selected word for further refinement does appear in the search string because if it was in the search string before, it would have been searched for.

Regarding claim 10, Mattson/Fein teaches a personal information manager comprising:

a microprocessor (see Mattson Abstract; “*speech processor*”);

memory operably connected to the microprocessor and storing a database (see Mattson column 4 lines 11-34; “*the system manager 16 may enable the speech*

processor and a text memory means 26 to transform the verbal observations to text or words and store words generated");

a data input device operably connected to the microprocessor and configured to receive an audio data stream and decode the received audio data stream into text (see Mattson column 4 lines 11-34; "*the physician's observations are converted from the spoken word to text*");

said database storing decoded text, and a table of explicit commands, each said explicit command being associated with at least one argument (see Mattson column 3 lines 34-50, "*A speech processor 12 analyzes the received audio signal and generates corresponding electronic words or text, as is known in the art. A command interpreter 14 compares each generated word or text with a list of preselected control commands or command words. The list of preselected control commands includes all or some of the commands which the operator had previously been able to enter on the keyboard C.*");

a dialog manager module executed by the microprocessor, said dialog manager comparing said decoded text received from said data input device with said table of explicit commands and assigning a match score to the decoded text, said dialog manager module ranks the match score of the decoded text and if a match score assigned to a given word of decoded text is greater than a threshold score then a user intention is judged recognized, and the dialog manager module assigns values determined from the decoded text to the at least one arguments associated with the explicit command (Fein, addressed below; see also Dunning, addressed below);

an information storage/retrieval module executed by the microprocessor and storing and retrieving text to/from said database in response to explicit commands received from the dialog manager (see Mattson column 4 lines 11-34, "*The audio signal may also be conveyed directly to a voice recorder 24, preferably a voice actuated voice recorder. In this manner, observations and volume imager commands made by the physician during the medical procedure are recorded on the voice recorder. This recording can later be used as input to the speech processor. Alternately, the verbal observations may be recorded on the voice track of the video recorder 20. As yet another option, the system manager 16 may enable the speech processor and a text memory means 26 to transform the verbal observations to text or words and store words generated*"); and

an output module converting text received from said dialog manager module into speech and outputting said speech in response to a data processing request (see Mattson column 5 lines 3-23; "*The speech synthesizer 42 is connected with the memory means 22 to convert the words of the stored patient history into audible signals to be supplied to a speaker 44*").

Mattson does not teach every limitation of a dialog manager module executed by the microprocessor, said dialog manager comparing said decoded text received from said data input device with said table of explicit commands and assigning a match score to the decoded text, said dialog manager module ranks the match score of the decoded text and if a match score assigned to a given word of decoded text is greater than a threshold score then a user intention is judged recognized, and the dialog manager

module assigns values determined from the decoded text to the at least one arguments associated with the explicit command. Fein teaches a method and system that performs implicit grammar checking of stored text during periods of microprocessor inactivity (see Fein column 4 lines 48-67, "*The preferred program includes a grammar checker program module. During idle periods, i.e., while a user is not entering data or commands, the document is automatically grammar checked in the background. The user does not have to enter any commands to initiate the background grammar check.*"; see Fein column 9 lines 19-29, "*The executable file 200 further includes an idle task manager 220 which manages tasks performed in the background. "Background" in this context means that the preferred application program module is idle, such as when the user is not typing and when no other commands are being executed. During an idle period, the idle task manager determines which idle tasks need to be performed based on a priority scheme. Those skilled in the art will be familiar with the operations and tasks performed by an idle task manager, including managing the order of performance of idle tasks during an idle period based on a priority scheme.*"; see also Fein column 16 lines 4-14, "*After marking the sentence with a grammarclean flag at step 535 or marking the characters comprising an error with a grammarerror flag at step 540, the method proceeds to decision step 545. A determination is made whether there are any more grammardirty flags or out-of-date grammarerror flags in the text run at decision step 545. If it is determined that there are no more grammardirty flags or out-of-date grammarerror flags in the text run being checked, then the method ends at step 517 and the preferred application program receives instruction from the idle task manager as to*

the next idle task to perform."). It would have been obvious to one having ordinary skill in the art at the time the invention was made to perform implicit tasks on the stored text during periods of microprocessor inactivity as taught by Fein in the invention of Mattson so that processing or analysis may be performed on the stored text without affecting the performance of the system.

Neither Mattson nor Fein teaches assigning a match score to the decoded text, said dialog manager module ranks the match score of the decoded text and if a match score assigned to a given word of decoded text is greater than a threshold score then a user intention is judged recognized, and the dialog manager module assigns values determined from the decoded text to the at least one arguments associated with the explicit command. Mattson teaches comparing the digitized signal or word with a library of stored digital words (see Mattson column 4 lines 35-54), but does not explicitly disclose how this is done internally. Dunning explicitly teaches determining a match score to determine if an input matches a stored value (see Dunning column 9 lines 7-13; "*Once an index has been built it can be used to identify an unknown signal. The unknown signal can also be broken into documents, quantized, and grouped into words. In one embodiment of the present invention, the words in the unknown documents can be compared to the words in the known documents in order to find a match and identify an unknown piece of music by its tag information.*"). It would have been obvious to one having ordinary skill in the art at the time the invention was made to compare input to stored values as taught by Dunning in the invention of Mattson/Fein in order to properly recognize if spoken commands match reserved words stored in the database.

Claims 13, 16, and 23-25 are rejected under 35 U.S.C. 103(a) as being unpatentable over Mattson (US 5,303,148), and Kotamarti, United States Patent Application Publication 2001/0043234.

Regarding claim 13, Mattson substantially teaches a personal information manager comprising:

a microprocessor (see Mattson Abstract; "*speech processor*");

memory operably connected to the microprocessor and storing a database (see Mattson column 4 lines 11-34; "*the system manager 16 may enable the speech processor and a text memory means 26 to transform the verbal observations to text or words and store words generated*");

a data input device operably connected to the microprocessor and configured to receive an audio data stream and decode the received audio data stream into text (see Mattson column 4 lines 11-34; "*the physician's observations are converted from the spoken word to text*");

said database storing decoded text, and a table of explicit commands, each said explicit command being associated at least one argument (see Mattson column 3 lines 34-50, "*A speech processor 12 analyzes the received audio signal and generates corresponding electronic words or text, as is known in the art. A command interpreter 14 compares each generated word or text with a list of preselected control commands or command words. The list of preselected control commands includes all or some of the commands which the operator had previously been able to enter on the keyboard C.*");

a dialog manager module executed by the microprocessor, said dialog manager comparing said decoded text received from said data input device with said table of explicit commands, if said decoded text matches one of the explicit commands, said dialog manager will assign values determined from the decoded text to the at least one arguments associated with the explicit command (Kotamarti, addressed below);

an information storage/retrieval module executed by the microprocessor storing and retrieving text to/from said database in response to explicit commands received from the dialog manager (see Mattson column 4 lines 11-34, "*The audio signal may also be conveyed directly to a voice recorder 24, preferably a voice actuated voice recorder. In this manner, observations and volume imager commands made by the physician during the medical procedure are recorded on the voice recorder. This recording can later be used as input to the speech processor. Alternately, the verbal observations may be recorded on the voice track of the video recorder 20. As yet another option, the system manager 16 may enable the speech processor and a text memory means 26 to transform the verbal observations to text or words and store words generated*"); and

an output module converting text received from said dialog manager module into speech and outputting said speech in response to a data processing request (see Mattson column 5 lines 3-23; "*The speech synthesizer 42 is connected with the memory means 22 to convert the words of the stored patient history into audible signals to be supplied to a speaker 44*"),

wherein if the argument specified in said explicit command corresponds to a unique entry in the database then the information storage/retrieval module will perform

the actions associated with the explicit command, otherwise the information storage/retrieval module will instruct the dialog manager module that the argument is ambiguous and the dialog manager module pass a prompt for clarification of the argument to the output module (Kotamarti, addressed below).

Mattson does not teach every limitation of a dialog manager module executed by the microprocessor, said dialog manager comparing said decoded text received from said data input device with said table of explicit commands, if said decoded text matches one of the explicit commands, said dialog manager will assign values determined from the decoded text to the at least one arguments associated with the explicit command. Mattson teaches the dialog manager comparing decoded text with a list of explicit commands, but is silent regarding assigning values from the decoded text to arguments associated with the explicit command. Kotamarti teaches a voice command system that assigns parameter values from the decoded text to arguments associated with the explicit command (see Kotamarti paragraph [0095], "*converting speech received from the user into user input strings for Metabrowser 750*"; see also Kotamarti paragraph [0140], "*The metabrowser receives a user input string and stores the string for subsequent submission. The string is submitted as a value for the form field identified by id. The strings may be a series of letters spelling words values for the field.*"). It would have been obvious to one having ordinary skill in the art at the time the invention was made to receive voice input specifying an explicit command and arguments as taught by Kotamarti in the invention of Mattson so that more specific commands may be given to the system.

Mattson also does not teach that the dialog manager module passes a prompt for clarification of ambiguous arguments to the output module. Kotamarti teaches prompting the user for clarification of ambiguous arguments (see Kotamarti paragraph [0146], "*This command allows definition of a secondary prompt that may be used when a user fails to provide appropriate user input.*"). It would have been obvious to one having ordinary skill in the art at the time the invention was made to prompt the user for clarification when the command is ambiguous as taught by Kotamarti in the invention of Mattson so that ambiguous commands may be resolved.

Regarding claim 16, Mattson/Kotamarti teaches that the decoded text is stored as a semi-structured collection of atoms (see Mattson column 4 lines 11-34, "*The audio signal may also be conveyed directly to a voice recorder 24, preferably a voice actuated voice recorder. In this manner, observations and volume imager commands made by the physician during the medical procedure are recorded on the voice recorder. This recording can later be used as input to the speech processor. Alternately, the verbal observations may be recorded on the voice track of the video recorder 20. As yet another option, the system manager 16 may enable the speech processor and a text memory means 26 to transform the verbal observations to text or words and store words generated. In this manner, the physician's observations are converted from the spoken word to text.*").

Regarding claim 23, Mattson/Kotamarti teaches that if the information storage/retrieval module judges that the argument is ambiguous it will pass characteristic words selected from data records corresponding to the argument to the

dialog manager module, and said dialog manager module instructs said output module to prompt the user to select a given said characteristic word used to refine the data processing request (see Kotamarti paragraph [0146], "*This command allows definition of a secondary prompt that may be used when a user fails to provide appropriate user input.*").

Regarding claim 24, Mattson/Kotamarti teaches a global word table containing a list of all the words stored in the database (see Mattson column 3 lines 34-50, "*A command interpreter 14 compares each generated word or text with a list of preselected control commands or command words. The list of preselected control commands includes all or some of the commands which the operator had previously been able to enter on the keyboard C.*"); and

said dialog manager module examining decoded text received from said data input device to determine whether it matches to a given said word in said global word table (see Mattson column 3 lines 34-50, "*A command interpreter 14 compares each generated word or text with a list of preselected control commands or command words. The list of preselected control commands includes all or some of the commands which the operator had previously been able to enter on the keyboard C. In response to receiving each of the preselected command words, the command interpreter generates the same control signal that the keyboard c produces for the corresponding command. The control input signal is processed by the system manager 16 which is responsible for activating the appropriate response of the whole system.*");

wherein a request to prompt the user for clarification is queued if the decoded text does not match any word in said global word table (see Kotamarti paragraph [0146], "*Prompt s, where s:=<<string>.vertline.<@<fn>".wav- ">>. This command allows definition of a secondary prompt that may be used when a user fails to provide appropriate user input. The user may provide user input that is not appropriate in a variety of ways. For example, a user may fail to respond after a period of time, may provide an unrecognized user command (e.g. depress a key pad on a phone with no corresponding preset), or may provide user input that does not correspond to a value identifier when providing input for an enumerated field.*").

Regarding claim 25, Mattson/Kotamarti teaches a local word table in said database (see Mattson column 3 lines 34-50, "*A command interpreter 14 compares each generated word or text with a list of preselected control commands or command words. The list of preselected control commands includes all or some of the commands which the operator had previously been able to enter on the keyboard C.*");

said information storage/retrieval module stores the decoded text as atoms of data in the database, each said atom having a unique identifier (see Mattson column 3 lines 34-50, "*A command interpreter 14 compares each generated word or text with a list of preselected control commands or command words. The list of preselected control commands includes all or some of the commands which the operator had previously been able to enter on the keyboard C. In response to receiving each of the preselected command words, the command interpreter generates the same control signal that the keyboard c produces for the corresponding command. The control input signal is*

processed by the system manager 16 which is responsible for activating the appropriate response of the whole system."); and

said local word table containing a list of words contained in each atom of data and the number of times each word appears in a given atom (see Mattson column 3 lines 34-50);

wherein if a number of atoms matching a data retrieval request exceeds a predetermined number, said dialog manager module prompts a user to select a given characteristic word from a list of characteristic words, said characteristic words being derived from the local word tables of atoms matching said data retrieval request, said selected characteristic word being appended to a search string of the data retrieval request, thereby reducing the number of atoms matching a data retrieval request (see Kotamarti paragraph [0140], "*GET p id [fname], where p:=<str>.vertline.<@<fn>".wav">.vertline.[<@<fn.txt>>]<string>. The Get command causes the metabrowser to play the prompt specified by p. The parameter P may specify a string or a wav file. The metabrowser receives a user input string and stores the string for subsequent submission. The string is submitted as a value for the form field identified by id. The strings may be a series of letters spelling words values for the field.*").

Claims 14, 17, and 18 are rejected under 35 U.S.C. 103(a) as being unpatentable over Mattson (US 5,303,148) *supra*, Kotamarti (US 2001/0043234) *supra*, and Gould et al. (hereinafter Gould), United States Patent 6,839,669.

Regarding claim 14, Mattson/Kotamarti teaches every limitation of claim 14 except that the database further includes at least one dedicated database selected from the list Calendar and Contacts, said Calendar database storing at least one of Dates and Times of appointments, and said Contacts database storing address information. Gould teaches a device equipped with speech recognition software for automatically performing tasks relating to calendars or contacts lists (see Gould column 2 lines 8-26, "*First, a user turns on a recorder and states what he wants to have happen. For example, the user might say "schedule an appointment with Joel for tomorrow at 3 o'clock" or "send an email to Paula Paula, please review the following . . . " or "take down a note I just met with Mr. Smith and the result of the meeting . . . ". When the user returns to his office, or is otherwise able to access his computer, he connects the recorder to his computer and clicks one button to have the system automatically transcribe the recorded information. The user then reviews the transcription and clicks a second button to instruct the computer to perform the actions represented by the transcription. The computer then automatically sends the email, schedules the appointment, adds the notes to the appropriate contact records, and performs any other necessary action. Thus, the techniques provide the user with a portable, pocket-sized assistant that uses speech recognition software and a contact manager or similar product to make actions happen.*"). It would have been obvious to one having ordinary skill in the art at the time the invention was made to update calendars and/or contacts using speech recognition as taught by Gould in the invention of Mattson/Kotamarti in

order to provide more functionality to users and because the inventions are in the same field of endeavor.

Regarding claim 17, Mattson/Kotamarti/Gould teaches that data is stored/retrieved to/from the at least one dedicated database in response to explicit commands (see Mattson column 4 lines 11-34, "*The audio signal may also be conveyed directly to a voice recorder 24, preferably a voice actuated voice recorder. In this manner, observations and volume imager commands made by the physician during the medical procedure are recorded on the voice recorder. This recording can later be used as input to the speech processor. Alternately, the verbal observations may be recorded on the voice track of the video recorder 20. As yet another option, the system manager 16 may enable the speech processor and a text memory means 26 to transform the verbal observations to text or words and store words generated*").

Regarding claim 18, Mattson/Kotamarti/Gould teaches that said dialog manager module has a record mode and a dialog mode, in said record mode said dialog manager module comparing said decoded text received with said table of explicit commands to determine whether it contains an explicit command, if said decoded text does not match one of the explicit commands then the said dialog manager stores the decoded text in said database (see Mattson column 3 lines 34-50, "*A command interpreter 14 compares each generated word or text with a list of preselected control commands or command words. The list of preselected control commands includes all or some of the commands which the operator had previously been able to enter on the keyboard C. In response to receiving each of the preselected command words, the*

command interpreter generates the same control signal that the keyboard c produces for the corresponding command."; see also Mattson column 4 lines 11-34, "*the verbal observations may be recorded on the voice track of the video recorder 20. As yet another option, the system manager 16 may enable the speech processor and a text memory means 26 to transform the verbal observations to text or words and store words generated. In this manner, the physician's observations are converted from the spoken word to text.*").

Claim 15 is rejected under 35 U.S.C. 103(a) as being unpatentable over Mattson (US 5,303,148) *supra*, Kotamarti (US 2001/0043234) *supra*, Gould (US 6,839,669) *supra*, and Maekawa et al. (hereinafter Maekawa), United States Patent Application Publication 2004/0068406.

Regarding claim 15, Mattson/Kotamarti/Gould teaches every limitation of claim 15 except that one or more explicit commands may be associated with two or more spoken outputs, wherein the output module randomly selects one of the spoken outputs associated with a given explicit command. Maekawa teaches a speech synthesis system that randomly selects replies to output to the user (see Maekawa paragraph [0148], "*More specifically, any reply is randomly selected from the plurality of replies corresponding to the above category data which are retained in the conversation database 354, and the selected reply is output. (Alternately, replies may be selected such that the same reply is not selected in succession.) It should be noted that it is not always necessary to retain a plurality of replies as described above. However, if an*

appropriate number of replies are sustained and selection among the replies is performed randomly, it is easier to provide a conversation with a natural impression."). It would have been obvious to one having ordinary skill in the art at the time the invention was made to randomly select replies to output to the user as taught by Maekawa in the invention of Mattson/Kotamarti/Gould in order to provide a conversation with a natural impression.

Claims 19 and 20 are rejected under 35 U.S.C. 103(a) as being unpatentable over Mattson (US 5,303,148) *supra*, Kotamarti (US 2001/0043234) *supra*, Gould (US 6,839,669) *supra*, and Fein (US 6,012,075) *supra*.

Regarding claim 19, Mattson/Kotamarti/Gould teaches every limitation of claim 19, but does not fully disclose that said dialog manager module examines text stored in said database to determine whether the stored text contains implicit data processing requests, where the implicit data processing requests are determined by examining a semantic class of the text, said dialog manager module adding implicit processing requests to an implicit processing queue and executing implicit processing requests during periods of microprocessor inactivity. Fein teaches a method and system that performs implicit grammar checking of stored text during periods of microprocessor inactivity (see Fein column 4 lines 48-67, "*The preferred program includes a grammar checker program module. During idle periods, i.e., while a user is not entering data or commands, the document is automatically grammar checked in the background. The user does not have to enter any commands to initiate the background grammar check.*";

see Fein column 9 lines 19-29, "*The executable file 200 further includes an idle task manager 220 which manages tasks performed in the background. "Background" in this context means that the preferred application program module is idle, such as when the user is not typing and when no other commands are being executed. During an idle period, the idle task manager determines which idle tasks need to be performed based on a priority scheme. Those skilled in the art will be familiar with the operations and tasks performed by an idle task manager, including managing the order of performance of idle tasks during an idle period based on a priority scheme.*"; see also Fein column 16 lines 4-14, "*After marking the sentence with a grammarclean flag at step 535 or marking the characters comprising an error with a grammarerror flag at step 540, the method proceeds to decision step 545. A determination is made whether there are any more grammardirty flags or out-of-date grammarerror flags in the text run at decision step 545. If it is determined that there are no more grammardirty flags or out-of-date grammarerror flags in the text run being checked, then the method ends at step 517 and the preferred application program receives instruction from the idle task manager as to the next idle task to perform.*"). It would have been obvious to one having ordinary skill in the art at the time the invention was made to perform implicit tasks on the stored text during periods of microprocessor inactivity as taught by Fein in the invention of Mattson/Kotamarti/Gould so that processing or analysis may be performed on the stored text without affecting the performance of the system.

Regarding claim 20, Mattson/Kotamarti/Gould/Fein teaches that data is stored in the at least one dedicated database in response to implicit data processing requests

(see Mattson column 4 lines 11-34, "*The audio signal may also be conveyed directly to a voice recorder 24, preferably a voice actuated voice recorder. In this manner, observations and volume imager commands made by the physician during the medical procedure are recorded on the voice recorder. This recording can later be used as input to the speech processor. Alternately, the verbal observations may be recorded on the voice track of the video recorder 20. As yet another option, the system manager 16 may enable the speech processor and a text memory means 26 to transform the verbal observations to text or words and store words generated*").

Claims 21 and 22 are rejected under 35 U.S.C. 103(a) as being unpatentable over Mattson (US 5,303,148), Kotamarti (US 2001/0043234) *supra*, and Dunning (US 7,162,482) *supra*.

Regarding claim 21, Mattson/Kotamarti teaches every limitation of claim 21 except that if said decoded text matches more than one explicit command in said explicit command table then said dialog manager module assigns a match score to each explicit commands and selects said explicit command having a highest match score as said explicit command. Dunning explicitly teaches determining a match score to determine if an input matches a stored value (see Dunning column 9 lines 7-13; "*Once an index has been built it can be used to identify an unknown signal. The unknown signal can also be broken into documents, quantized, and grouped into words. In one embodiment of the present invention, the words in the unknown documents can be compared to the words in the known documents in order to find a match and identify*

an unknown piece of music by its tag information."). It would have been obvious to one having ordinary skill in the art at the time the invention was made to compare input to stored values as taught by Dunning in the invention of Mattson/Kotamarti in order to properly recognize if spoken commands match reserved words stored in the database.

Regarding claim 22, Mattson/Kotamarti/Dunning teaches that if said highest match score is less than a threshold score then said dialog manager module passes an instruction to said output module to prompt the user to select a given explicit command from among a selected number of said explicit commands (see Kotamarti paragraph [0146], "*Prompt s, where s:=<<string>.vertline.<@<fn>".wav- ">>. This command allows definition of a secondary prompt that may be used when a user fails to provide appropriate user input. The user may provide user input that is not appropriate in a variety of ways. For example, a user may fail to respond after a period of time, may provide an unrecognized user command (e.g. depress a key pad on a phone with no corresponding preset), or may provide user input that does not correspond to a value identifier when providing input for an enumerated field.*").

Claim 26 is rejected under 35 U.S.C. 103(a) as being unpatentable over Mattson (US 5,303,148) *supra*, Kotamarti (US 2001/0043234) *supra*, and Flinchem et al. (hereinafter Flinchem), United States Patent 6,307,548.

Regarding claim 26, Mattson/Kotamarti teaches every limitation of claim 26 except that said characteristic words are derived by selecting a predetermined number of the most frequently occurring words from the local word tables of the atoms matching

a data retrieval request, provided that the selected word does not already appear in the search string of the data retrieval request. Flinchem teaches deriving characteristic words by selecting the most frequently occurring words from the database matching the data retrieval request (see Flinchem column 15 lines 37-52, "*One of the key features of the preferred vocabulary module tree data structure is that the objects associated with each node are stored in the node data structure 400 according to their frequency of use. That is, the object constructed by the first instruction in packet 406 has a higher frequency of use than that constructed by the second instruction (if present) in 406, which has a higher frequency of use than the third instruction (if present). In this manner, the objects are automatically placed in the object list so that they are sorted according to decreasing frequency of use. For purposes of this description, frequency of use of a word object refers to the likelihood of using a given word within a representative corpus of use, which is proportional to the number of times that each word occurs in the corpus. In the case of word stem objects, frequency of use is determined by summing the frequencies of all words which share the same stem.*"). It would have been obvious to one having ordinary skill in the art at the time the invention was made to derive characteristic words by frequency of occurrence as taught by Flinchem in the invention of Mattson/Kotamarti so that characteristic words can be queried more quickly.

Claims 27 and 38-41 are rejected under 35 U.S.C. 103(a) as being unpatentable over Mattson (US 5,303,148), Fein (US 6,012,075) *supra*, and Dunning (US 7,162,482) *supra*, and Kotamarti (US 2001/0043234) *supra*.

Regarding claim 27, Mattson/Fein/Dunning teaches every limitation of claim 27 except that if the argument specified in said explicit command corresponds to a unique entry in the database then the information storage/retrieval module will perform the actions associated with the explicit command, otherwise the information storage/retrieval module will instruct the dialog manager module that the argument is ambiguous and the dialog manager module pass a prompt for clarification of the argument to the output module. Kotamarti teaches prompting the user for clarification of ambiguous arguments (see Kotamarti paragraph [0146], "*This command allows definition of a secondary prompt that may be used when a user fails to provide appropriate user input.*"). It would have been obvious to one having ordinary skill in the art at the time the invention was made to prompt the user for clarification when the command is ambiguous as taught by Kotamarti in the invention of Mattson/Fein/Dunning so that ambiguous commands may be resolved.

Regarding claim 38, Mattson/Fein/Dunning/Kotamarti teaches that if said highest match score is less than a threshold score then said dialog manager module passes an instruction to said output module to prompt the user to select a given explicit command from among a selected number of said explicit commands (see Kotamarti paragraph [0146], "*Prompt s, where s:=<<string>.vertline.<@<fn>".wav- ">>. This command allows definition of a secondary prompt that may be used when a user fails to*

provide appropriate user input. The user may provide user input that is not appropriate in a variety of ways. For example, a user may fail to respond after a period of time, may provide an unrecognized user command (e.g. depress a key pad on a phone with no corresponding preset), or may provide user input that does not correspond to a value identifier when providing input for an enumerated field.").

Regarding claim 39, Mattson/Fein/Dunning/Kotamarti teaches that if the information storage/retrieval module judges that the argument is ambiguous it will pass characteristic words selected from data records corresponding to the argument to the dialog manager module, and said dialog manager module instructs said output module to prompt the user to select a given said characteristic word used to refine the data processing request (see Kotamarti paragraph [0146], "*This command allows definition of a secondary prompt that may be used when a user fails to provide appropriate user input.*").

Regarding claim 40, Mattson/Fein/Dunning/Kotamarti teaches a global word table containing a list of all of the words stored in the database (see Mattson column 3 lines 34-50, "*A command interpreter 14 compares each generated word or text with a list of preselected control commands or command words. The list of preselected control commands includes all or some of the commands which the operator had previously been able to enter on the keyboard C.*"); and

said dialog manager module examining decoded text received from said data input device to determine whether it matches to a given said word in said global word table (see Mattson column 3 lines 34-50, "*A command interpreter 14 compares each*

generated word or text with a list of preselected control commands or command words. The list of preselected control commands includes all or some of the commands which the operator had previously been able to enter on the keyboard C. In response to receiving each of the preselected command words, the command interpreter generates the same control signal that the keyboard c produces for the corresponding command. The control input signal is processed by the system manager 16 which is responsible for activating the appropriate response of the whole system.");

wherein a request to prompt the user for clarification is queued if the decoded text does not match any word in said global word table (see Kotamarti paragraph [0146], "*Prompt s, where s:=<<string>.vertline.<@<fn>".wav- ">>. This command allows definition of a secondary prompt that may be used when a user fails to provide appropriate user input. The user may provide user input that is not appropriate in a variety of ways. For example, a user may fail to respond after a period of time, may provide an unrecognized user command (e.g. depress a key pad on a phone with no corresponding preset), or may provide user input that does not correspond to a value identifier when providing input for an enumerated field.*").

Regarding claim 41, Mattson/Fein/Dunning/Kotamarti teaches a local word table in said database;

said information storage/retrieval module stores the decoded text as atoms of data in the database, each said atom having a unique identifier (see Mattson column 3 lines 34-50, "*A command interpreter 14 compares each generated word or text with a list of preselected control commands or command words. The list of preselected control*

commands includes all or some of the commands which the operator had previously been able to enter on the keyboard C."); and

said local word table containing a list of words contained in each atom of data and the number of times each word appears in a given atom (see Mattson column 3 lines 34-50, "*A command interpreter 14 compares each generated word or text with a list of preselected control commands or command words. The list of preselected control commands includes all or some of the commands which the operator had previously been able to enter on the keyboard C. In response to receiving each of the preselected command words, the command interpreter generates the same control signal that the keyboard c produces for the corresponding command. The control input signal is processed by the system manager 16 which is responsible for activating the appropriate response of the whole system.*");

wherein if a number of atoms matching a data retrieval request exceeds a predetermined number, said dialog manager module prompts a user to select a given characteristic word from a list of characteristic words, said characteristic words being derived form the local word tables of atoms matching said data retrieval request, said selected characteristic word being appended to a search string of the data retrieval request, thereby reducing the number of atoms matching a data retrieval request (see Kotamarti paragraph [0140], "*GET p id [fname], where p:=<str>.vertline.<@<fn>".wa-v">.vertline.[<@<fn.txt>>]<string>. The Get command causes the metabrowser to play the prompt specified by p. The parameter P may specify a string or a wav file. The metabrowser receives a user input string and stores the string for subsequent*

submission. The string is submitted as a value for the form field identified by id. The strings may be a series of letters spelling words values for the field.").

Claims 28, 29, 30, and 33 are rejected under 35 U.S.C. 103(a) as being unpatentable over Mattson (US 5,303,148), Fein (US 6,012,075) *supra*, and Dunning (US 7,162,482) *supra*, Kotamarti (US 2001/0043234) *supra*, and Maekawa (US 2004/0068406) *supra*.

Regarding claim 28, Mattson/Fein/Dunning/Kotamarti teaches every limitation of claim 28 except that each explicit command is associated with at least one message identifier. Maekawa teaches a speech synthesis system that randomly selects replies to output to the user, each command associated with at least one reply (see Maekawa paragraph [0148], "*More specifically, any reply is randomly selected from the plurality of replies corresponding to the above category data which are retained in the conversation database 354, and the selected reply is output. (Alternately, replies may be selected such that the same reply is not selected in succession.) It should be noted that it is not always necessary to retain a plurality of replies as described above. However, if an appropriate number of replies are sustained and selection among the replies is performed randomly, it is easier to provide a conversation with a natural impression.*"). It would have been obvious to one having ordinary skill in the art at the time the invention was made to have replies associated with each command as taught by Maekawa in the invention of Mattson/Fein/Dunning/Kotamarti in order to provide a conversation with a natural impression.

Regarding claim 29, Mattson/Fein/Dunning/Kotamarti/Maekawa teaches that two or more explicit commands are associated with a given message identifier (see Maekawa paragraph [0148], "*More specifically, any reply is randomly selected from the plurality of replies corresponding to the above category data which are retained in the conversation database 354, and the selected reply is output. (Alternately, replies may be selected such that the same reply is not selected in succession.) It should be noted that it is not always necessary to retain a plurality of replies as described above. However, if an appropriate number of replies are sustained and selection among the replies is performed randomly, it is easier to provide a conversation with a natural impression.*").

Regarding claim 30, Mattson/Fein/Dunning/Kotamarti/Maekawa teaches that each explicit command comprises at least one command word, and different explicit command words may be associated with a given message identifier (see Maekawa paragraph [0148], "*More specifically, any reply is randomly selected from the plurality of replies corresponding to the above category data which are retained in the conversation database 354, and the selected reply is output. (Alternately, replies may be selected such that the same reply is not selected in succession.) It should be noted that it is not always necessary to retain a plurality of replies as described above. However, if an appropriate number of replies are sustained and selection among the replies is performed randomly, it is easier to provide a conversation with a natural impression.*").

Regarding claim 33, Mattson/Fein/Dunning/Kotamarti teaches every limitation of claim 33 except that each explicit command is associated with at least one spoken

outputs, wherein the output module randomly selects one of the spoken outputs associated with a given explicit command. Maekawa teaches a speech synthesis system that randomly selects replies to output to the user (see Maekawa paragraph [0148], "*More specifically, any reply is randomly selected from the plurality of replies corresponding to the above category data which are retained in the conversation database 354, and the selected reply is output. (Alternately, replies may be selected such that the same reply is not selected in succession.) It should be noted that it is not always necessary to retain a plurality of replies as described above. However, if an appropriate number of replies are sustained and selection among the replies is performed randomly, it is easier to provide a conversation with a natural impression.*"). It would have been obvious to one having ordinary skill in the art at the time the invention was made to randomly select replies to output to the user as taught by Maekawa in the invention of Mattson/Fein/Dunning/Kotamarti in order to provide a conversation with a natural impression.

Claims 31 and 34-37 are rejected under 35 U.S.C. 103(a) as being unpatentable over Mattson (US 5,303,148), Fein (US 6,012,075) *supra*, and Dunning (US 7,162,482) *supra*, Kotamarti (US 2001/0043234) *supra*, and Gould (US 6,839,669) *supra*.

Regarding claim 31, Mattson/Fein/Dunning/Kotamarti teaches every limitation of claim 31 except that the database further includes at least one dedicated database selected from the list Calendar and Contacts, said Calendar database storing at least one of Dates and Times of appointments, and said Contacts database storing address

information. Gould teaches a device equipped with speech recognition software for automatically performing tasks relating to calendars or contacts lists (see Gould column 2 lines 8-26, "*First, a user turns on a recorder and states what he wants to have happen. For example, the user might say "schedule an appointment with Joel for tomorrow at 3 o'clock" or "send an email to Paula Paula, please review the following . . . " or "take down a note I just met with Mr. Smith and the result of the meeting . . . ". When the user returns to his office, or is otherwise able to access his computer, he connects the recorder to his computer and clicks one button to have the system automatically transcribe the recorded information. The user then reviews the transcription and clicks a second button to instruct the computer to perform the actions represented by the transcription. The computer then automatically sends the email, schedules the appointment, adds the notes to the appropriate contact records, and performs any other necessary action. Thus, the techniques provide the user with a portable, pocket-sized assistant that uses speech recognition software and a contact manager or similar product to make actions happen.*"). It would have been obvious to one having ordinary skill in the art at the time the invention was made to update calendars and/or contacts using speech recognition as taught by Gould in the invention of Mattson/Fein/Dunning/Kotamarti in order to provide more functionality to users and because the inventions are in the same field of endeavor.

Regarding claim 34, Mattson/Fein/Dunning/Kotamarti/Gould teaches that data is stored/retrieved to/from the at least one dedicated database in response to explicit commands (see Mattson column 4 lines 11-34, "*The audio signal may also be conveyed*

directly to a voice recorder 24, preferably a voice actuated voice recorder. In this manner, observations and volume imager commands made by the physician during the medical procedure are recorded on the voice recorder. This recording can later be used as input to the speech processor. Alternately, the verbal observations may be recorded on the voice track of the video recorder 20. As yet another option, the system manager 16 may enable the speech processor and a text memory means 26 to transform the verbal observations to text or words and store words generated").

Regarding claim 35, Mattson/Fein/Dunning/Kotamarti/Gould teaches that said dialog manager module has a record mode and a dialog mode, in said record mode said dialog manager module comparing said decoded text received with said table of explicit commands to determine whether it contains an explicit command, if said decoded text does not match one of the explicit commands then the said dialog manager stores the decoded text in said database (see Mattson column 3 lines 34-50, "*A command interpreter 14 compares each generated word or text with a list of preselected control commands or command words. The list of preselected control commands includes all or some of the commands which the operator had previously been able to enter on the keyboard C. In response to receiving each of the preselected command words, the command interpreter generates the same control signal that the keyboard c produces for the corresponding command.*"; see also Mattson column 4 lines 11-34, "*the verbal observations may be recorded on the voice track of the video recorder 20. As yet another option, the system manager 16 may enable the speech processor and a text memory means 26 to transform the verbal observations to text or*

words and store words generated. In this manner, the physician's observations are converted from the spoken word to text.").

Regarding claim 36, Mattson/Fein/Dunning/Kotamarti/Gould teaches that said dialog manager module examines text stored in said database to determine whether the stored text contains implicit data processing requests, where implicit data processing requests are determined by examining a semantic class of the text, said dialog manager module adding implicit processing requests to an implicit processing queue and executing implicit processing requests during periods of microprocessor inactivity (see Fein column 4 lines 48-67, "*The preferred program includes a grammar checker program module. During idle periods, i.e., while a user is not entering data or commands, the document is automatically grammar checked in the background. The user does not have to enter any commands to initiate the background grammar check.*"; see Fein column 9 lines 19-29, "*The executable file 200 further includes an idle task manager 220 which manages tasks performed in the background. "Background" in this context means that the preferred application program module is idle, such as when the user is not typing and when no other commands are being executed. During an idle period, the idle task manager determines which idle tasks need to be performed based on a priority scheme. Those skilled in the art will be familiar with the operations and tasks performed by an idle task manager, including managing the order of performance of idle tasks during an idle period based on a priority scheme.*"; see also Fein column 16 lines 4-14, "*After marking the sentence with a grammarclean flag at step 535 or marking the characters comprising an error with a grammarerror flag at step 540, the method*

proceeds to decision step 545. A determination is made whether there are any more grammardirty flags or out-of-date grammarerror flags in the text run at decision step 545. If it is determined that there are no more grammardirty flags or out-of-date grammarerror flags in the text run being checked, then the method ends at step 517 and the preferred application program receives instruction from the idle task manager as to the next idle task to perform.").

Regarding claim 37, Mattson/Fein/Dunning/Kotamarti/Gould teaches that data is stored in the at least one dedicated database in response to implicit data processing requests (see Mattson column 4 lines 11-34, "*The audio signal may also be conveyed directly to a voice recorder 24, preferably a voice actuated voice recorder. In this manner, observations and volume imager commands made by the physician during the medical procedure are recorded on the voice recorder. This recording can later be used as input to the speech processor. Alternately, the verbal observations may be recorded on the voice track of the video recorder 20. As yet another option, the system manager 16 may enable the speech processor and a text memory means 26 to transform the verbal observations to text or words and store words generated*").

Claim 32 is rejected under 35 U.S.C. 103(a) as being unpatentable over Mattson (US 5,303,148), Fein (US 6,012,075) *supra*, and Dunning (US 7,162,482) *supra*, and Maekawa (US 2004/0068406) *supra*.

Regarding claim 32, Mattson/Fein/Dunning teaches that each explicit command is associated with at least one spoken outputs, wherein the output module randomly

selects one of the spoken outputs associated with a given explicit command. Maekawa teaches a speech synthesis system that randomly selects replies to output to the user (see Maekawa paragraph [0148], "*More specifically, any reply is randomly selected from the plurality of replies corresponding to the above category data which are retained in the conversation database 354, and the selected reply is output. (Alternately, replies may be selected such that the same reply is not selected in succession.) It should be noted that it is not always necessary to retain a plurality of replies as described above. However, if an appropriate number of replies are sustained and selection among the replies is performed randomly, it is easier to provide a conversation with a natural impression.*"). It would have been obvious to one having ordinary skill in the art at the time the invention was made to randomly select replies to output to the user as taught by Maekawa in the invention of Mattson/Fein/Dunning in order to provide a conversation with a natural impression.

Claim 42 is rejected under 35 U.S.C. 103(a) as being unpatentable over Mattson (US 5,303,148), Fein (US 6,012,075) *supra*, and Dunning (US 7,162,482) *supra*, Kotamarti (US 2001/0043234) *supra*, and Flinchem (US 6,307,548) *supra*.

Regarding claim 42, Mattson/Fein/Dunning/Kotamarti teaches that said characteristic words are derived by selecting a predetermined number of the most frequently occurring words from the local word tables of the atoms matching a data retrieval request, provided that the selected word does not already appear in the search string of the data retrieval request. Flinchem teaches deriving characteristic words by

selecting the most frequently occurring words from the database matching the data retrieval request (see Flinchem column 15 lines 37-52, "*One of the key features of the preferred vocabulary module tree data structure is that the objects associated with each node are stored in the node data structure 400 according to their frequency of use. That is, the object constructed by the first instruction in packet 406 has a higher frequency of use than that constructed by the second instruction (if present) in 406, which has a higher frequency of use than the third instruction (if present). In this manner, the objects are automatically placed in the object list so that they are sorted according to decreasing frequency of use. For purposes of this description, frequency of use of a word object refers to the likelihood of using a given word within a representative corpus of use, which is proportional to the number of times that each word occurs in the corpus. In the case of word stem objects, frequency of use is determined by summing the frequencies of all words which share the same stem.*"). It would have been obvious to one having ordinary skill in the art at the time the invention was made to derive characteristic words by frequency of occurrence as taught by Flinchem in the invention of Mattson/Fein/Dunning/Kotamarti so that characteristic words can be queried more quickly.

Response to Arguments

Applicant's arguments with respect to the claims have been considered but are moot in view of the new ground(s) of rejection.

Conclusion

The prior art made of record and not relied upon is considered pertinent to applicant's disclosure:

- Grayson et al. (US 2004/0001065) Electronic conference program
- Dostie et al. (US 2004/0021691) Method, system and media for entering data in a personal computing device
- Comer et al. (US RE39326 E) Method and apparatus for suggesting completions for a partially entered data item based on previously-entered, associated data items

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Stephen Alvesteffer whose telephone number is (571)270-1295. The examiner can normally be reached on Monday-Friday 9:30AM-6:00PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner’s supervisor, William Bashore can be reached on (571)272-4088. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Stephen Alvesteffer
Examiner
Art Unit 2175

/S. A./
Examiner, Art Unit 2175

/William L. Bashore/
Supervisory Patent Examiner, Art Unit 2175